Enabling Intelligence at Network Edge: An Overview of Federated Learning

Howard H. YANG[1], ZHAO Zhongyuan[2], Tony Q. S. QUEK[1]

(1. Singapore University of Technology and Design, Singapore 487372, Singapore;
2. Beijing University of Post and Telecommunication, Beijing 100876, China)

Abstract: The burgeoning advances in machine learning and wireless technologies are forging a new paradigm for future networks, which are expected to possess higher degrees of intelligence via the inference from vast dataset and being able to respond to local events in a timely manner. Due to the sheer volume of data generated by end-user devices, as well as the increasing concerns about sharing private information, a new branch of machine learning models, namely federated learning, has emerged from the intersection of artificial intelligence and edge computing. In contrast to conventional machine learning methods, federated learning brings the models directly to the device for training, where only the resultant parameters shall be sent to the edge servers. The local copies of the model on the devices bring along great advantages of eliminating network latency and preserving data privacy. Nevertheless, to make federated learning possible, one needs to tackle new challenges that require a fundamental departure from standard methods designed for distributed optimizations. In this paper, we aim to deliver a comprehensive introduction of federated learning. Specifically, we first survey the basis of federated learning, including its learning structure and the distinct features from conventional machine learning models. We then enumerate several critical issues associated with the deployment of federated learning in a wireless network, and show why and how technologies should be jointly integrated to facilitate the full implementation from different perspectives, ranging from algorithmic design, on-device training, to communication resource management. Finally, we conclude by shedding light on some potential applications and future trends.

Keywords: federated learning; edge intelligence; learning algorithm; communication efficiency; privacy and security

DOI: 10.12142/ZTECOM.202002002

http://kns.cnki.net/kcms/detail/34.1294.TN.20200610.1007.002.html, published online June 10, 2020

Manuscript received: 2020-02-10

Citation (IEEE Format): H. H. Yang, Z. Y. Zhao, and T. Q. S. Quek, "Enabling intelligence at network edge: an overview of federated learning," *ZTE Communications*, vol. 18, no. 2, pp. 02 - 10, Jun. 2020. doi: 10.12142/ZTECOM.202002002.

1 Introduction

The networking system is experiencing a paradigm shift from a conventional cloud computing architecture that aggregates the computational resources at a data center, to mobile edge systems which largely deploy computational power to the network edges to meet the demands from mobile applications—which are most thriving today—and support resource-constrained nodes reachable only over unreliable network connections[1]. Moreover, along with the burgeoning progress of machine learning research, it is expect-

ed that by integrating machine learning algorithms to the edge nodes, future networks will be able to utilize local data to conduct intelligent inference and control on many activities, e.g., learning activities of mobile phone users, predicting health events from wearable devices, or detecting burglaries within smart homes[2].

However, as the data is usually generated at the end-user devices, the sheer volume of the dataset as well as the rising concerns about sharing private information often makes the users reluctant to send their raw data to the edge server for the training of any model—even that can eventually benefit them in return. In response to this dilemma, a new machine learning model has emerged, namely federated learning, that allows decoupling of data acquisition and computation at the central unit[3–5]. Specifically, rather than collecting all the data to a central unit for training, federated learning brings the models directly to the end-user devices for training, where only the resultant parameters shall be sent to the edge servers that reside in an edge node. This salient feature of on-device training brings along great advantages of eliminating the large communication overheads as well as preserving data privacy, and hence making federated learning particularly relevant for mobile applications. These properties also identify the federated learning as one of the most promising factors to an intelligent mobile edge network[6–9].

Nevertheless, in order to deliver a successful deployment of federated learning, one also needs to tackle new challenges that require a fundamental departure from the standard methods designed for distributed optimization[3],[10]. Particularly, unlike many traditional machine learning models, where an algorithm runs on a large dataset partitioned homogeneously across multiple servers in the cloud, the federated learning often operates in a mobile edge system, in which a server orchestrates the training with a union of end-user devices, which have non independent and identically distributed (i.i.d.) and unbalanced dataset, and communicate over a resource-limited spectrum[11–12]. In that regard, the staleness becomes more paramount to the training process[13] and security issues also arise that make the learning architecture vulnerable[14]. Addressing these issues requires joint studies from many aspects, including the learning algorithm, system design, and communication and information theory[15–16]. In response, Ref. [10] discussed the possible directions to improve the training efficiency when encountering with heterogeneous datasets. Moreover, Ref. [6] investigated the end-to-end latency, reliability, and scalability of a federated learning empowered edge network. In the particular context of deep learning, Ref. [8] explored the challenges and approaches to integrate the learning algorithm into network edge via a federated approach; Ref. [9] discussed a number of guidelines for the implementation of federated learning with the wireless channels. With these efforts, the results are fruitful: As will be detailed in Section 4, there are numerous applications that can benefit a lot by adopting federated learning. To that end, the central thrust of this paper is to deliver a comprehensive introduction to the federated learning system as well as to appeal for more research devoted into this emerging field. It is also noteworthy that while a few surveys on the topic of federated learning have been now available, our work puts a particular focus on the integration of the wireless infrastructures (such as the mobile edge network) as a supporting platform and the federated learning as an operation system, which ultimately achieves the network intelligence by jointly running them together.

The remainder of this paper is organized as follows. In Section 2, we introduce the basic structure and the defining characteristics of a federated learning model. The techniques to the core of a practical implementation of the federated learning system are elaborated in Section 3. Section 4 discusses the potential applications and future trends of federated learning, followed by the conclusion remarks in Section 5.

2 Federated Learning: Basis and Properties

In this section, we detail the basic architecture of a federated learning model running on the mobile edge system. A number of key features associated with such a setting will also be presented.

2.1 Basic Architecture

As illustrated in **Fig. 1**[17], the network elements involved in the federated learning include a central unit, e.g., the edge server that resides at a base station or access point and a number of end-user devices, in which they collaboratively learn a statistical model. The model is typically devised by a model engineer for a particular application, with which the server then orchestrates the training process with the end-user devices by repeating the following steps[3–4].

1) Client selection: The server selects from a subset of its clients, namely the end-user devices, which meet the eligibility requirements, e.g., mobile phones or tablets that currently have a wireless connection, for one round of training.

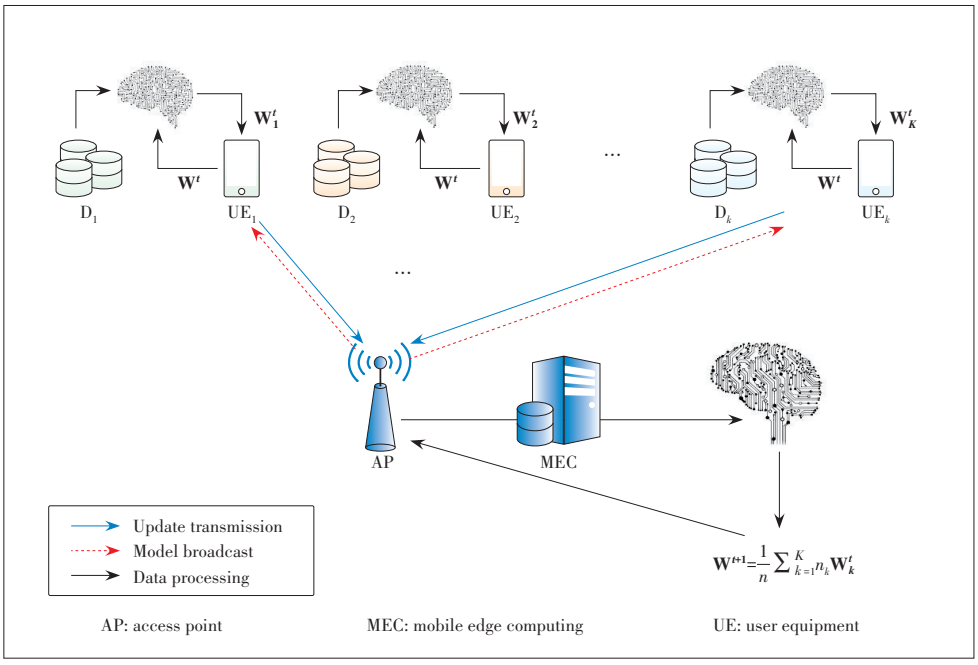
2) Broadcast: The selected clients download the current model, including the weights and a training program, from the server for local computing.

3) End-user computation: Each selected device performs a local computation, usually in the form of stochastic gradient descend (SGD), for a given period, and uploads the resultant parameters to the server.

4) Update aggregation: The server collects the updates from the end-user devices—in the form of either trained parameters or gradients—and aggregates, in general by a weighted average, the collected results.

5) Model update: The server locally updates the shared model based on the aggregated update computed from the clients that participated in the current round.

After a sufficient number of training and update exchanges



▲**Figure 1. Illustration of the network architecture, in which a mobile edge system is integrated with federated learning.**

(usually termed as communication rounds) between the server and the clients, the global statistical model is able to converge to its optimal and the end users can benefit from a collaboratively learned model.

1) The advantage: By training via federated learning, end users are able to directly download the model, perform computing on the devices, and send back the resultant trained parameters; in this way, the end users decouple the necessity of sharing local data and hence reserves privacy. Additionally, the local training also abbreviates the upload of raw data, which can be very large in size and consume a lot of energy for the upload. To that end, the federated learning is particularly relevant to wireless applications.

2) The challenge: The potential drawback of federated learning is also obvious. As the training is at a large scale amongst heterogeneous entities, e.g., different end terminals can have various processing power and communication conditions, the learning efficiency can be much lower than that in a data center. On top of this, the communication is often unreliable in the federated learning environment and security issue is more paramount under such a setting.

In the sequel, we will point out the possible directions to overcome the crux and finally realize the potential of federated learning. Before that, let us pause a while and clarify the most distinguishing features of such a learning model.

2.2 Distinguishing Features

At the first sight, it might seem that the federated learning is simply another format of distributed learning. These two machine learning models share several properties in common; for instance, the computing is carried out by a number of end terminals and the terminals iteratively collaborate via a central entity. However, there are many more features that distinguish the federated learning from those more conventional models. We highlight the key features of federated learning as follows.

• Non-i.i.d. dataset: The most distinct feature of federated learning is that the dataset of each end-user device is highly personalized and hence the dataset is usually non-i.i.d. across users. The sources of the dependence and non-identicalness are due to the fact that data collected at each device corresponds to a particular user, a particular geographic location, and/or a particular time period. As such, unlike situations in the conventional setup where the dataset is completely shuffled and i.i.d., in federated learning, the non-i.i.d. structure may lead to the local minimum of each device diverting from the global minimum, and requires a rethinking of learning model to take into account such differences in the process.

• Unbalanced data size: Aside from being non-i.i.d. distributed, the dataset of each end-user device also differs in size. Therefore, the training procedure at each end terminal can be highly unbalanced, because some terminals that have small datasets can complete the training in a short period of time, while those with large dataset sizes may take a longer time to complete the local training. Moreover, due to the unbalanced nature, some devices, e.g., those with a large dataset, may contribute more to the overall model than others, and hence how to account for such difference in the learning algorithm is also important.

• Limited communication resources: As the communications between end-user devices and a central entity often take place at the network edge, where spectrum is the medium to conduct communications, the transmissions are by nature unreliable. Moreover, as the wireless resources are usually limited, it is necessary to select the appropriate number of users each round for the communication. All these can impose more significant impact of staleness on the overall training efficiency.

• Privacy/security issues: Whilst learning under the federated setting abbreviates the sharing of local data, it does not promise a perfect protection of privacy. In fact, one can still extract leaky information from upload parameters and retrieve the original information to an approximation extend[11]. Moreover, under the federated setting, the end-devices are more

vulnerable to malicious attacks in this case and it is easily for some adversary users to inject malicious information into the system.

Note that a marked property of many of the features/problems discussed above is that they are inherently interdisciplinary and solving them likely requires not just machine learning, but also techniques from distributed optimization, security, differential privacy, fairness, compressed sensing, systems, information theory, statistics and more. In fact, many of the hardest problems are at the intersections of these areas and hence a cross-area study/collaboration is essential to ongoing progress.

3 Towards Practical Implementation

As mentioned in the previous section, despite the potentials to endow the mobile edge network with a higher degree of intelligence via federated learning, it requires a full cooperation between computing and communication to realize the full potential of such a scheme. In this section, we elaborate several key aspects that we believe to be lying at the core of achieving the final goal.

3.1 Efficient Learning Algorithms

The primary factor to the implementation of federated learning is an efficient algorithm. Due to the non-i.i.d. nature of the dataset, a model training process of the federated learning can be very different from the conventional counterparts. In particular, unlike scenarios under the distributed computing, where each end terminal possesses a statistically identical model (namely the empirical loss function), in the federated learning, each end-user device can have very different empirical loss due to the personalized dataset. As such, the local minimum may differ from the global minimum and the learning algorithm shall be reengineered to account for this fact[10]. Besides, as the communication resource is limited, the edge server can only choose a subset of users for the update in each round of communication. Therefore, how to select users appropriately also plays a critical role in the overall learning efficiency[12].

3.1.1 Optimization and Model Aggregation

Because of the non-i.i.d. nature of user dataset, treating all samples equivalently at the global model may not make a solid sense. Therefore, how to craft a more appropriate objective function is an important aspect to research. Besides, the current state-of-the-art training is mostly SGD-base, which is well-known for slow converging. Therefore, how to develop more effective algorithm will also determine the efficiency of federated learning. Moreover, owing to the vast number, each device is likely to participate only a few rounds in the training of a global model, so stateless algorithms are necessary to investigate.

In the aggregation stage, the common approach is the Federated Averaging algorithm, an adaption of parallel SGD that takes a weighted average of the collected parameters according to their dataset size. While the effectiveness of such an approach has been demonstrated in different models, it is still unknown whether this is the optimal way of aggregating parameters and further investigation is necessary.

3.1.2 Sampling and Client Selection

Due to the unbalanced structure of datasets as well as the limited bandwidth, the sampling, of not just the data points for computing but also the clients to conduct local trainings in each communication round, plays a critical role that determines the overall learning efficiency. In particular, as each end-user device may correspond to a specific local minimum of empirical loss, spending a lot of time on the local training may bear the risk of leading the parameters to diverge from the global minimum. On the other hand, as the global communication can take up a much longer period than the local computing, it is also desirable to reduce the communication rounds. As such, how to strike a balance between local computing and global communication is important to the efficiency of federated learning. In response, it is suggested that the sampling data size of each local training shall be adaptively adjusted across the global learning period.

On top of the sampling of dataset for local training, in the global aggregation stage, the edge server can only select a portion of users out of the total due to the limited bandwidth available. Therefore, for the client, i.e., end-user device, selection is also critical for the performance of federated learning. In the context of mobile edge system, it has been shown that by taking the channel quality into consideration and selecting the end-user devices with the best channel qualities, the learning efficiency can be effectively boosted up[12], as demonstrated in **Fig. 2**. Besides, it is also important to take into account the staleness and the significance of updates in the client selection stage[17].

3.2 Model Compression

Although the processing power of mobile devices has surged over the last decade by the hardware revolution, these terminals are still subject to power and storage constraints, making it problematic to deploy the federated learning toward a deep and large scale. The difficulty mainly attributes to two reasons. One is that a deep neural network often consists of an abundant amount of activation units and interconnecting links, and hence training such a model will inevitably incur excessive energy consumption and, if not worse, memory occupation. The other is that, even the task of model training can be accomplished at the user side, sending the resultant parameters, which are generally high dimension vectors, to the server requires not just high transmit power but also wide mobile spectrum, which imposes very high communication cost. Nonetheless, this does not mean one has no hope to adopt the most

fruitful achievement of machine learning, namely the deep neural network, in the federated setup. Two powerful approaches shed the light for overcoming the setbacks:

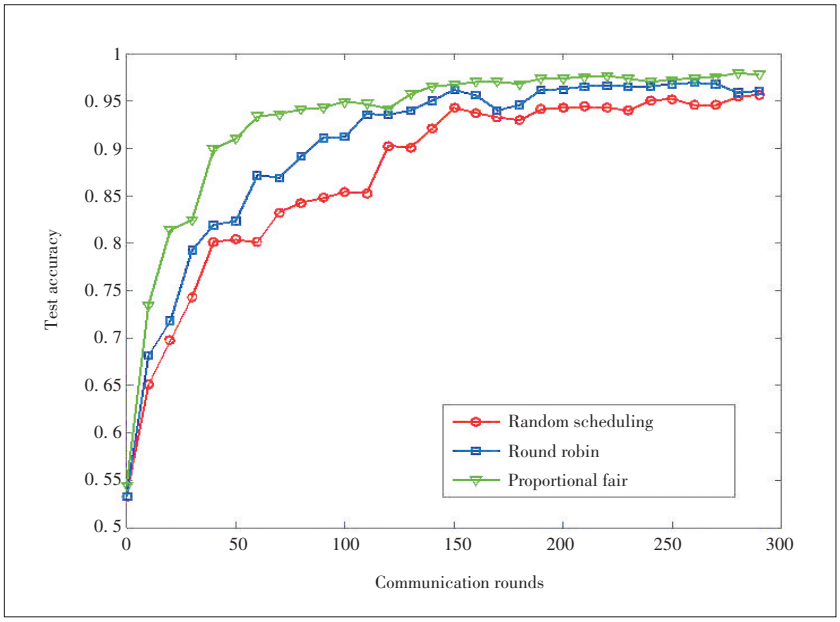
1) Architecture compression: This approach aims to save the cost from the computing perspective of neural network via pruning the connecting links and shrinking the size of the network[7]. The idea of link pruning stems from the fact that the majority of links connecting different layers of neurons are usually associated with very small weights. In order words, the most effective component of a neural network is architecturally sparse. Therefore, it is feasible to mute a number of links that have small weights—so as to skimp on the caching memory—without affecting the overall accuracy. Moreover, despite the unprecedented success brought by deep learning, there are many applications in which using a small neural network is able to achieve as good the performance as a large one. As such, directly reducing the size of neural network at the user side is also an appropriate choice to attain marked savings in both energy and memory consumption.

2) Gradient compression: This approach tackles the issue from the perspective of communication, by trading the estimation accuracy for better communication efficiency. In particular, by noticing that practical applications of machine learning often do not require very high accuracy, one can compress the high-dimension trained gradients (which can include millions of coefficients) into low dimension surrogates via different levels of quantization[18 - 19]. As a result, the packet size to encapsulate the trained results can be significantly reduced, which not only saves the radiated power at each device, but also facilitates the decoding process at the server. It is noteworthy that to balance the tradeoff between communication cost and training accuracy, the level of quantization shall be adapted to the particular location of a user. For instance, for users located in proximity to the edge node, they can conduct less quantization and maintain the high accuracy of the results, while for those located far away, they shall compress the trained results more aggressively in order to succeed the communication and engage in the training process.

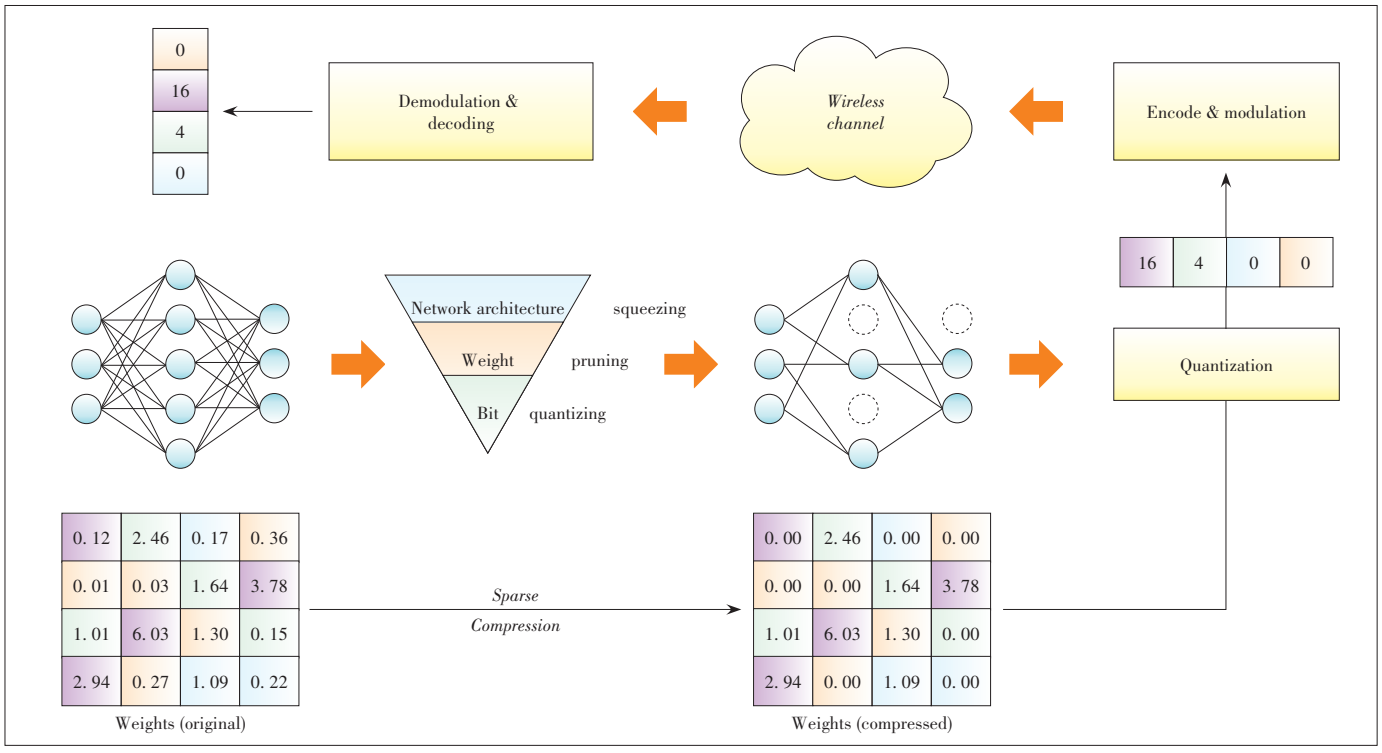
A complete process of model compression is illustrated in **Fig. 3**; we can see that it is feasible to remove a number of links with small weights in the neural network. Moreover, some neurons with only a few connections can also be muted. The architecture compression can thus transform the learning model into a sparse version, which can achieve almost the same performance as the original neural network. Another part is associated with the gradient compression, as the generated forms of parameters are often continuous with long digits, which are not suitable for the transmissions via wireless channels. By using appropriate quantization methods, the data volume of the update results can be significantly reduced, which not only saves the power consumption of end-user devices, but also facilitates the decoding procedure at the server side. To mitigate the impact of quantization noise, sophisticated parameter strategies are also necessary to minimize the model accuracy loss. It is worthwhile to mention that due to potential failure and retransmissions, the weights before and after the encoding/decoding process may appear in different orders. Nonetheless, the server can still leverage the sequential number to rearrange the weights before the global aggregation.

3.3 Advanced Communication and Networking Techniques

It has become a consensus that the communication efficiency is also one of the first-order concerns of federated learning, particularly due to the fact that the training involves a vast number of end-user device communications through a limited wireless bandwidth. In that respect, the technologies that enhance the spectral efficiency can be a critical solution to this dilemma. Specifically, the development of new technologies, e. g., the massive multiple input multiple output (MIMO), full duplex or non-orthogonal multiple access (NOMA), that are able to support more channel accesses over the same bandwidth will facilitate the deployment of federated learning. For instance, by deploying an excessive number of antennas at the base station, multiple devices can be simultaneously selected for parameter update in each round of communication, which, as demonstrated by a number of literatures, can help accelerate the convergence of federated learning algorithm. In a similar spirit, one can also leverage the techniques from full duplex or NOMA to increase the number of updates collectible in each global aggregation and hence speedup the



▲**Figure 2. Test accuracy of federated learning under different scheduling policies.**



▲**Figure 3. Basic flow of model compression in the federated learning system.**

training process. Besides, the ultra-reliable low latency communication (URLLC) that reduces the latency in the transmission is also a good candidate for more real-time learning tasks. A joint design that takes in the processing power and communication capability from both sides will also enhance the operation efficiency[15 – 16], [20 – 22].

Aside from communication efficiency, advanced networking technology is also important for the federated learning. In general, the federated learning involves a central server that orchestrates the training process and receives the contributions of end-user devices. Being as a central player, the server also represents a potential point of great failure[10]. As such, even though large companies or organizations can take this role in certain applications, a reliable and powerful central server may not always be available in more collaborative learning scenarios. Moreover, the server may even become a bottleneck when the number of clients is very large. To that end, it is suggested to replace communication with the server by a more distributed manner, namely peer-to-peer communication between individual devices. For that reason, advanced device-to-device (D2D) communication and interference management schemes can be a dominant factor to the overall performance. The self-organized networking techniques may have significant influence on the performance.

3.4 Privacy Preserving Technologies

Despite the raw data is not explicitly shared in the context of federated learning, it is still possible for adversaries to retrieve the original information to an approximation extend, especially when the learning architecture and parameters are not completely protected. In fact, due to the share nature of wireless medium, the intermediate results such as parameter update from an optimization algorithm are exposed during the transmission, which may leak out private information. Moreover, the existence of malevolent users may incur further security issues. Therefore, the design of federated learning into a mobile edge system needs further protection of parameters as well as investigations on the tradeoffs between the privacy security-level and the system performance[23].

In the federated learning process, there exists several fatal points that have privacy and security issues. We enumerate them into the following categories[14].

3.4.1 Privacy Protection at User Side

In a federated learning algorithm, end users need to iteratively upload their learning results to the edge server for global aggregation, but these users may not trust the server since a curious entity might take a look at the uploaded parameter to infer the underlined information. To address this concern, the end users can employ some privacy-preservation technologies as follows.

1) Perturbation: The idea of perturbation is adding noise to the uploaded parameters by clients. This line of work often uses differential privacy[24] to obscure certain sensitive attributes until the third party is not able to distinguish the individual, thereby making the data impossible to be restored so as to protect user privacy.

2) Dummy: The concept of dummy method stems from the

location privacy protection. By sending dummy model parameters along with the true one to the server, the end users can thus hide their contribution during training. Because of the aggregation processed at the server, the system performance can still be guaranteed.

3.4.2 Privacy Protection at Server Side

After collecting the updated parameters from the end-user devices, the server generally performs a weighted average to produce a new model. However, when the server broadcasts the aggregated parameters back to the users, the information may leak out as there may exist eavesdroppers. Thus, protections at the server side are also of significance.

1) Privacy-enabled aggregation: While the general purpose of aggregation at the server side is to produce an improved learning model, it is possible to scramble parameters before aggregating or enlarging the set of collected clients, which can prevent the adversaries or untrusted server from inspecting client information according to the aggregated parameters.

2) Secure multi-party computation (SMC): The central idea of SMC is to use encryption to increase protection of user updates, instead of only revealing the sum after a sufficient number of updates. Specifically, SMC is a four-round interactive protocol optionally enabled during the reporting phase of a given communication round. In each protocol round, the server gathers messages from all devices, and then uses the set of device messages to compute an independent response and return to each device. The third round constitutes a commit phase, during which devices upload cryptographically masked model updates to the server. Finally, there is a finalization phase during which devices reveal sufficient cryptographic secrets to allow the server to unmask the aggregated model update.

3.4.3 Security Protection for Learning Framework

This aspect mainly considers the model stealing attacks. In particular, any participant in the training process may introduce hidden backdoor functionality into the global model, e.g., to ensure that an image classifier assigns an attacker-chosen label to images with certain features, or that a word predictor completes certain sentences with an attacker. Consequently, there are also some protecting measures on the security design for this.

1) Homomorphic encryption: Homomorphic encryption aims to protect the parameter exchange process via encryption mechanism, by means of encoding the parameters before upload, and to transmit along with the public-private decoding keys for the intended entity to decipher.

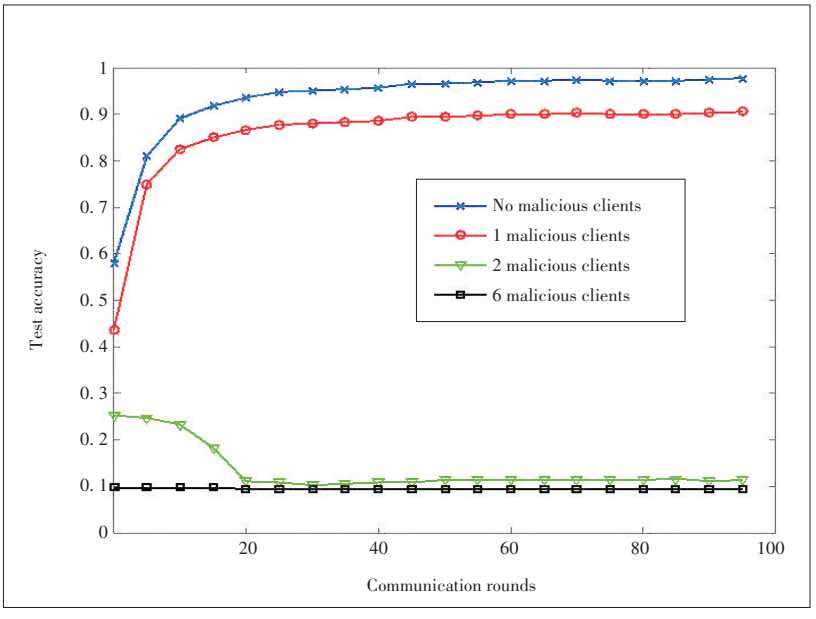
2) Back-door defender: This is a crucial issue with the federated learning, as a malicious user may act as an innocent user but injecting certain parameters to pollute the global parameter. In consequence, other end-user devices may encounter severe malfunctioning and breakdown. Therefore, effective approaches shall be developed to protect the users from these attacks.

In order to illustrate the impact of malicious attacks on the performance of federated learning, we carry out an experiment (**Fig. 4**[14]). Particularly, a convolutional neural network (CNN) is set up with 30 end-user devices participated in, whereas the malicious clients will upload fake values of parameters in each communication round. It can be seen that the system performance can significantly curtail by malicious attacks, and even enter a breakdown when there are too many malicious clients participating in. As such, security is of significant to the performance of federated learning.

While we have listed out several concerns on the implementation of federated learning and the approaches to address these issues, another important practical consideration for federated learning is the composability of these methods. The schemes of tackling each of these aspects shall not be devised in isolation but need to be combined with each other. For instance, the efficient learning algorithm will need to be designed in consideration of learning efficiency as well as privacy preserving. Also, the model compression shall also be contended with privacy preserving.

4 Potential Applications and Future Trends

The future trends of mobile edge networks are to integrate the supply and demand of services, being able to identify a particular application to the network and respond promptly. By employing the federated learning as an operational system to the network architecture, a more intelligent network system can be foreseen in the future[2].



▲**Figure 4. Performance of federated learning with different malicious users.**

From the perspective of network architecture, the federated learning can be integrated with content caching and edge computing at the edge of a mobile network to reduce backhaul traffic loads. The general idea of caching at the network edge is, when availed with a priori information of each individual preference distributions, to optimally place the desired content resource in the edge server so as to respond to user request more swiftly. It thus simultaneously enhances the energy efficiency, reduces the service latency, and relieves the backhaul load. Despite such benefits, the gain from caching alone is only pronounced when the users' preference distributions are a priori and highly homogeneous, i.e., the users tend to request the same contents. These two constraints, however, are less likely to be satisfied in next-generation wireless applications that possess a higher degree of heterogeneity. On one hand, the users' preference distributions vary drastically across time and space, thus making them extremely difficult to be estimated and tracked, especially when the number of mobile devices becomes large. On the other hand, in practice, the users' preference distributions are highly diverse due to the personality differences. Therefore, conventional model-base designs may not be suitable for such a task because it is not capable of considering multitude of factors that influence content popularity. Moreover, directly accessing the privacy-sensitive user data for content differentiation may not be possible in practice. Federated learning with the premise of utilizing the locally trained models rather than directly accessing the user data seems to be a match made in heaven for content popularity prediction in proactive caching in wireless networks. For instance, in augmented reality (AR), federated learning can be used to learn certain popular elements of the augmentations from the other users without obtaining their privacy-sensitive data directly. This popular information is then pre-fetched and stored locally to reduce the latency.

From the perspective of resource management[6], the federated learning paradigm can be used to improve the spectrum sensing efficiency, and thus flexible and adaptive sharing and reuse strategies can be implemented to the communication system. Apart from the radio access, the next-generation network needs to deal with more volatile traffic conditions. Along with the warp speed of progress of mobile applications, different types of traffic, which may be bursty, long-lasting, or with short packet size, coexist in the network. Consequently, centralized strategies, where information about traffic pattern is gathered in the database of a server to infer the circumstance, may not always be appropriate. Therefore, the future of network traffic management will be dependent on the decentralized training approaches such as the federated learning. In this context, the on-device training can provide more real-time reaction to schedule the traffic of the most appropriate users. A specific instance of application is the coexistence of dedicated short-range communication and cellular-connected vehicle-to-everything in the same intelligent transport systems.

Finally, from the perspective of end user applications, federated learning is expected to find many landing grounds. For instance, by equipping sensors with federated learning algorithms, one can construct a local Internet-of-Things (IoT) network with intelligent monitoring system that can quickly identify certain events and quickly respond to them. Hospitals, if endowed with a federated learning system for disease monitoring, might increase the doctors' intention to share information and prevent certain catastrophe in the early stage. In the area of retailing, the federated learning system can leverage data from a wild range of entities to increase the accuracy of prediction on demands, and thus help providers/owners prepare supplies in a proper manner. In self-driving cars, information related to traffic can be learned through vehicles on the road using federated learning and stored in the roadside units, which facilitates the efficiency of an autonomous driving operation system.

Notably, a number of future studies immediately follow from the above discussions. For instance, one can investigate how to adopt the federated learning to inference the distribution of local demand so as to provide appropriate guidance on the allocation of caching contents on the network edge that can reduce communication burden. In the context of mobile resource management, how to leverage the federated learning to extract the individual traffic distributions to further benefit the allocation of global spectral resources is also a concrete direction. To sum up, the integration of federated learning and mobile edge network can provide a unified platform to support a variety of applications, and we also advocate for subsequent studies to build up the federated intelligence ecosystem.

5 Conclusions

In this paper, we provided an overview to the federated learning system. Specifically, we elaborated the basic architecture of the federated learning model and the salient features, in particular the non-i.i.d. and unbalanced dataset, unreliable and limited communication resource, as well as privacy and security issues, that distinguish it from the conventional ones. Furthermore, we presented a number of practical approaches that enable the implementation of federated learning into a mobile edge system. Among them, we emphasized the importance from aspects of algorithm design, model compression and communication efficiency. Lastly, we presented several applications that are most foreseeable to benefit from applying federated learning. In summary, we believe that federated learning is one of the building blocks in achieving an intelligent network and we expect that more interesting research issues will appear in this area.

References

[1] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: the communication perspective [J]. IEEE communications surveys & tutorials, 2017, 19(4): 2322 – 2358. DOI: 10.1109/comst.2017.2745201
[2] LETAIEF K B, CHEN W, SHI Y M, et al. The roadmap to 6G: AI empowered wireless networks [J]. IEEE communications magazine, 2019, 57(8): 84 – 90
[3] KONEČNÝ J, MCMAHAN H B, YU F, et al. Federated learning: strategies for improving communication efficiency [EB/OL]. (2016-10-18) [2019-09-17]. https://arxiv.org/abs/1610.05492
[4] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [EB/OL]. (2016-02-17) [2019-09-17]. https://arxiv.org/abs/1602.05629
[5] SMITH V, FORTE S, MA C X, et al. CoCoA: a general framework for communication-efficient distributed optimization [J]. Journal of machine learning research, 2018, 18(230): 1 – 49
[6] PARK J, SAMARAKOON S, BENNIS M, et al. Wireless network intelligence at the edge [J]. Proceedings of the IEEE, 2019, 107(11): 2204 – 2239. DOI: 10.1109/jproc.2019.2941458
[7] ZHAO Z Y, FENG C Y, YANG H H, et al. Federated-learning-enabled intelligent fog radio access networks: fundamental theory, key techniques, and future trends [J]. IEEE wireless communications, 2020, 27(2): 22 – 28. DOI: 10.1109/mwc.001.1900370
[8] ZHOU Z, CHEN X, LI E, et al. Edge intelligence: paving the last mile of artificial intelligence with edge computing [J]. Proceedings of the IEEE, 2019, 107(8): 1738 – 1762. DOI: 10.1109/jproc.2019.2918951
[9] ZHU G X, LIU D Z, DU Y Q, et al. Toward an intelligent edge: wireless communication meets machine learning [J]. IEEE communications magazine, 2020, 58(1): 19 – 25. DOI: 10.1109/mcom.001.1900103
[10] KAIROUZ P, MCMAHAN H B, AVENTET B, et al. Advances and open problems in federated learning [EB/OL]. (2019-12-10) [2019-09-17]. https://arxiv.org/abs/1912.04977
[11] WANG S Q, TUOR T, SALONIDIS T, et al. Adaptive federated learning in resource constrained edge computing systems [J]. IEEE journal on selected areas in communications, 2019, 37(6): 1205 – 1221
[12] YANG H H, LIU Z, QUEK T Q S, et al. Scheduling policies for federated learning in wireless networks [J]. IEEE transactions on communications, 2020, 68(1): 317 – 333
[13] DAI W, ZHOU Y, DONG N Qet al. Toward understanding the impact of staleness in distributed machine learning [C]//International Conference for Learning Representations (ICLR). New Orleans, Louisiana, 2019: 1 – 6
[14] MA C, LI J, DING M, et al. On safeguarding privacy and security in the framework of federated learning [J]. IEEE network, 2020: 1 – 7. DOI: 10.1109/mnet.001.1900506
[15] TRAN N H, BAO W, ZOMAYA A, et al. Federated learning over wireless networks: optimization model design and analysis [C]//IEEE Conference on Computer Communications (INFOCOM). Paris, France, 2019. DOI: 10.1109/infocom.2019.8737464
[16] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks [EB/OL]. [2019-09-17]. https://arxiv.org/pdf/1909.07972
[17] YANG H H, ARAFA A, QUEK T Q S, et al. Age-based scheduling policy for federated learning in mobile edge networks [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Barcelona, Spain, 2020. DOI: 10.1109/icassp40776.2020.9053740
[18] DU Y Q, YANG S, HUANG K B. High-dimensional stochastic gradient quantization for communication-efficient edge learning [J]. IEEE transactions on signal processing, 2020, 68: 2128 – 2142.
[19] ZHU G X, DU Y Q, GÜNDÜZ D, et al. One-bit over-the-air aggregation for communication-efficient federated edge learning: design and convergence analysis [EB/OL]. [2020-01-16]. https://arxiv.org/pdf/2001.05713
[20] ZHU G X, WANG Y, HUANG K B. Broadband analog aggregation for low-latency federated edge learning [J]. IEEE transactions on wireless communications, 2020, 19(1): 491 – 506. DOI: 10.1109/twc.2019.2946245
[21] YANG K, JIANG T, SHI Y M, et al. Federated learning via over-the-air computation [J]. IEEE transactions on wireless communications, 2020, 19(3): 2022 – 2035. DOI: 10.1109/twc.2019.2961673
[22] AMIRI M M, GUNDUZ D. Machine learning at the wireless edge: distributed stochastic gradient descent over-the-air [J]. IEEE transactions on signal processing, 2020, 68: 2155 – 2169
[23] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning via additively homomorphic encryption [J]. IEEE transactions on information forensics and security, 2018, 13(5): 1333 – 1345. DOI: 10.1109/tifs.2017.2787987
[24] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis [M]//Theory of cryptography. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2006: 265 – 284. DOI: 10.1007/11681878_14

Biographies

Howard H. YANG received the B.Sc. degree in communication engineering from Harbin Institute of Technology (HIT), China, in 2012, the M.Sc. degree in electronic engineering from Hong Kong University of Science and Technology (HKUST), China, in 2013, and the Ph.D. degree in electronic engineering from Singapore University of Technology and Design (SUTD), Singapore, in 2017. His background also features appointments at the University of Texas at Austin, USA and Princeton University, USA. His research interests cover various aspects of wireless communications, networking and signal processing, currently focusing on the modeling of modern wireless networks, high dimensional statistics, graph signal processing and machine learning. He received the IEEE WCSP 10-Year Anniversary Excellent Paper Award in 2019 and the IEEE WCSP Best Paper Award in 2014.

ZHAO Zhongyuan (zyzhao@bupt.edu.cn) received the B.S. and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), China, in 2009 and 2014, respectively. He is currently an associate professor with BUPT. His research interests include mobile cloud and fog computing and network edge intelligence. Dr. ZHAO serves as an editor of *IEEE Communications Letters* (since 2016). He was the recipient of the Best Paper Awards at the IEEE CIT 2014 and WASA 2015. He was also the recipient of Exemplary Reviewers-2017 of *IEEE Transactions on Communications*, and Exemplary Editor Award 2017 and 2018 of *IEEE Communication Letters*.

Tony Q. S. QUEK received the B.E. and M.E. degrees in electrical and electronics engineering from Tokyo Institute of Technology, Japan. At MIT, USA, he earned the Ph.D. in electrical engineering and computer science. Currently, he is the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD). He also serves as the acting head of Information System Technology and Design (ISTD) Pillar, sector lead for SUTD AI Program, and the deputy director of SUTD-ZJU IDEA. He is currently serving as an editor for the *IEEE Transactions on Wireless Communications*, the chair of IEEE VTS Technical Committee on Deep Learning for Wireless Communications as well as an elected member of the IEEE Signal Processing Society SPCOM Technical Committee. He received the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, 2017 CTTC Early Achievement Award, 2017 IEEE ComSoc AP Outstanding Paper Award, and 2016-2019 Clarivate Analytics Highly Cited Researcher. He is a Distinguished Lecturer of the IEEE Communications Society and a Fellow of IEEE.